目录

表格和插图清单

表格

插图

针对不平衡数据下二元分类问题的数据挖掘体系——银行电话营销策略研究

摘要

背景 近年来，不平衡数据分类是数据挖掘的重要研究方向之一。不平衡特性指的是数据集中各类别在数量上相差悬殊，比例高达 10:1、100:1 甚至更大。传统分类算法在应用到不平衡数据上时，会出现少数类的分类准确率远远低于多数类的分类准确率，而在许多实际应用中，往往少数类才是我们的关注对象。

目的 本文以对银行电话营销策略的研究为例，同时从数据层面和算法层面入手，旨在建立一套以数据为驱动的不平衡数据分类方法体系。

方法 首先，我们在了解数据经济背景和不平衡特性之后，确立了贯穿全文的评价二元分类器的综合指标 F-Score。在数据预处理之后，我们综合了 RFE、Boruta 和 XGBoost 三种先进的特征选择方法的结果，通过投票筛选出重要特征。

然后，基于已有的各种欠采样和过采样方法的优缺点，采用本文 5 种联合抽样方法对不平衡样本分别处理，并将其结果作用于随机森林、BP 神经网络、LightGBM、SVM 和 Logistic 回归等 11 个 R 开源的主流分类器。

最后，借助深度学习思想，我们创新性地将 5 个联合抽样方法下的共 55 个分类器结果作为样本新特征，用 XGBoost-Stacking 集成学习，再进一步地使用“分簇排序策略”选择性集成，从而得到最终的机器学习模型。

结果与意义 在我们的方法体系下所构建的机器学习模型，针对不平衡数据下的二元分类问题，查准率和召回率都非常高，甚至优于 Paulo Cortez 等前辈在相似数据集上的研究结果。不平衡数据的分类问题存在于各个领域，如疾病诊断、信用卡风险、图像异常识别、挖掘潜在客户、预测重大设备的故障、检测石油泄漏等。因此，本文的方法体系推广性强，且具有很强的实际意义。

关键词 机器学习、二元分类、不平衡样本、欠采样和过采样、选择集成学习、XGBoost、电话营销

A Data-Mining System of Methods for Binary Classification Based on Imbalanced Data – Research on Bank Telemarketing Campaign

Abstract

Background Recently, the academia of data mining attaches great importance to the problem of classification based on imbalanced data. Imbalance refers to a large difference between the number of each category in a data set, generally causing the sample ratio to be 10:1, 100:1 or even higher. Traditional classification algorithm, when applied to imbalanced data, often leads to a low accuracy for minority class, far less than the accuracy for majority class. However, in many real-world applications, the minority class is what we are usually more concerned with.

Objective This paper aims to propose a data-driven system of methods (involving data processing and algorithm) for binary classification based on imbalanced data, taking for example the research on bank telemarketing campaign.

Methods First of all, by analyzing the data set's economic background and its nature of imbalance, we establish an appraisal index called F-Score for binary classifier throughout the paper. After data pre-processing, we implement the three state-of-the-art feature selection methods – RFE, Boruta and XGBoost. Their results are summarized to conclude final important features by voting.

Then, considering the pros and cons of existing undersampling and oversampling methods, we employ 5 combined sampling methods to process the original data set. Their results are input for training 11 mainstream classifiers with open source on R, such as Random Forest, BP Neural Network, LightGBM, SVM and Logistic Regression.

Next, with the thought of deep learning, we innovatively regard as new features the results of 55 base classifiers under the 5 sampling methods previously mentioned and input the integrated features for ensemble learning using XGBoost-Stacking. Furthermore, through selective ensemble learning with the help of clustering and sorting strategy, we obtain the final machine learning model.

Results and Significance The machine learning model derived from our system of methods shows high Precision and Recall, better than the result of Paulo Cortez who conducted a similar research on the same data set before. Our system of methods deserves to be generalized to a variety of fields, including disease diagnosis, transaction risk of credit card, abnormal image identification, pinpointing potential customer, equipment failure detection, oil spills warning, etc.

Key words: machine learning, binary classification, imbalanced data, undersampling and oversampling, selective ensemble learning, XGBoost, telemarketing

一、引言

（一）研究背景

在数据挖掘中，分类问题可谓最为常见。一系列经典的分类算法如决策树，神经网络，SVM 都已成功地应用到很多领域，然而，针对不平衡数据样本，经典分类器却会遇到很多问题。不平衡样本指的是，其中某种类别的数据远远少于另一类数据的数据集，其中数量多的称为多数类，少的称为少数类。一个典型的例子——使用检测数据诊断癌症病人，病人数量是非常少的，比例只有几十万甚至几百万分之一，远远小于正常人。由于传统分类器追求整体准确率最高，因此一般对多数类有较高的识别率，对少数类的识别率很差。换言之，分类器通常会几乎把所有的样本判为正常人，这样准确率可以达到 99%，但这是没有意义的，把所有癌症病人误判为正常人，耽误了治疗，代价是很巨大的。在很多领域中，少数类的分类准确率才是重点，而不是整体准确率。

不平衡数据的其实广泛存在于实际生活，如预测信用卡风险、客户流失预测、金融风险评级、网络攻击检测，重大故障检测，挖掘潜在用户,检测石油泄漏。因此，研究不平衡数据的分类方法体系具有重要的理论价值和现实意义。

（二）研究现状

近些年，针对不平衡数据分类问题，比较成熟的对策大体上可以分为三种：一类是抽样方法，对数据进行处理，使数据趋向平衡；另一类是算法改进，较常见的是代价敏感学习；还有一类是集成方法，利用多个弱分类器，集成为一个性能更好的强分类器。

数据重采样——在数据层面解决不平衡样本问题，它主要分为“欠抽样”和“过抽样”两种重采样方法。“欠抽样”方案有 Tomek Link Removal 方法[10]，邻域清理法（NCL）[11]和压缩最近邻法（CNN）[12]等。至于“过抽样”方案，最初是随机复制少数类样本，但存在较大的过拟合风险。而后来有 Chawla 等人提出 SMOTE (Synthetic Minority Oversampling Technique) 方法[13]，它是一个直至目前仍被广泛应用的过抽样方法。

分类器集成——在算法层面解决不平衡样本问题。尽管数据重采样提升了分类器的分类效果，但受限于单个分类器的劣势，可能泛化能力较差。集成学习[14]的基本思想是：训练多个基分类器，汇总所有基分类器的结果为最终模型。典型

的集成学习有 Bagging[38]和 Boosting[39]，利用基分类器的差异和优势进行互补，能够提高分类精度、增强泛化能力，还能适应多种数据分布。更进一步，随后的研究以及 2000 年周志华等人的文章[23]表明，通过恰当方法剔除部分基分类器，可提升分类效果，即选择性集成方法[15]。

（三）本文主要工作及创新之处

本文针对在各领域内广泛存在的不平衡样本分类问题，在对前辈工作深入学习的基础上，同时从数据层面和算法层面入手，原创性地完成了以下工作。

首先，依据本文的数据背景，我们确定了研究目标和与之对应的不平衡数据二元分类模型的综合评价指标 F-Score。我们采用了 RFE、Boruta 和 XGBoost 三种比较先进的特征选择方法共同筛选出重要特征，以提升后续分类器性能。

其次，在对不平衡样本的处理方面，我们发现经典的 SMOTE 方法可能导致分类器边界模糊。因此本文决定采用 Tomek Link Removal、NCL 和 K-means Undersampling 方法跟经典的 SMOTE 方法结合，使得分类边界明显且样本趋向平衡。本文的 5 种联合抽样方法（简写为 SK, SNT, SNK, STK, SNTK）在 11 个分类器上的分类效果都比经典的 SMOTE 更加优秀。

然后，我们对训练数据采用了几乎所有 R 开源的主流分类器，包括 2016 年 12 月微软刚在 R 开源的 LightGBM 算法、随机森林[30,31]、ID3 决策树[28]、C50 决策树[29]、BP 神经网络[32,33]、MLP 多层神经网络[34,35]、线性判别分析（LDA）[25]、KNN[23,24]、Logistic 回归[27]、朴素贝叶斯[26]、支持向量机（SVM）[36]和 XGBoost[5]共 12 个分类器，发现数据不平衡导致分类效果普遍较差。此外，我们参考 Bagging 的思想，将 SVM 改进成 multi-SVM，在分类准确率不变的同时，把训练时间从 100 分钟降低到了 2 分钟，极大地降低了算法的时间复杂度。

最后，得到 55 个经过联合抽样方法改进的基分类器后，采用本文的 XGBoost-Stacking 选择集成框架进行集成，发现分类效果比经典的集成方式更加卓越，同时也优于 Sérgio Moro 与 Paulo Cortez 等人对于该数据集研究后的分类效果[1,2]。改进其一在于，以往的集成都是使用所有分类器，而本文结合不平衡特性，由综合评价指标 F-Score 筛选基分类器，采用“分簇排序策略”的选择性集成方法。其二在于，以往的集成都是将各分类器的结果进行线性加权得到最终分类结果，而本文采用了非线性的、训练高效且准确的 XGBoost 作为第二层的学习器，将第一层分类器的分类结果作为额外特征，进行再学习，得到最终分类结果。

二、数据介绍与研究目标

（一）数据背景

在本论文中，我们所研究的数据集由一家葡萄牙的零售银行所采集并汇总，它可以通过 URL http://archive.ics.uci.edu/ml/datasets/Bank+Marketing 下载得到，公开可用。

这个数据集属于商业、金融领域，它一共包含了 45211 个客户的 17 个属性的信息，内容与该银行当时所进行的营销活动直接相关，完全基于该银行的客户经理给该银行的广大客户打电话所了解到的信息（16 个属性），以及事后银行系统内记录在案的结果——该客户最终是否办理了定期存款（1 个属性）。

该数据集的前 16 个属性是特征，被分成三类（分别详见表 1、表 2 和表 3）；该数据集的最后一个属性是类标签，也是该数据集就分类问题而言的目标变量：该客户是否办理了定期存款（是：1；否：0）。

第一类特征是客户的基本信息（表 1）。

表 1　第一类特征介绍

–	变量名	类型	基本描述	具体类别
1	age	数值变量	–	–
2	job	分类变量	职业类型	admin., unknown, unemployed, management, housemaid, entrepreneur, student, blue-collar, self-employed, retired, technician, services
3	marital	分类变量	婚姻状态	married, divorced, single; note: "divorced" means divorced or widowed
4	education	分类变量	–	unknown, primary, secondary, tertiary
5	default	二分类变量	是否有信贷违约	yes, no
6	balance	数值变量	年均存款数额（欧元）	–
7	housing	二分类变量	是否有住房贷款	yes, no
8	loan	二分类变量	是否有个人贷款	yes, no

第二类特征包含了在当次营销活动中最后一次电话沟通的信息。

表 2　第二类特征介绍

–	变量名	类型	基本描述	具体类别
9	contact	分类变量	沟通方式类型	unknown, telephone, cellular
10	day	数值变量	该月的最后联系日	–
11	month	分类变量	该年的最后联系月	jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
12	duration	数值变量	最后联系时长（秒）	–

第三类特征包含了先前营销活动和当次营销活动的相关信息。

表 3　第三类特征介绍

–	变量名	类型	基本描述	具体类别
13	campaign	数值变量	在当次营销活动中被联系的次数	–
14	pdays	数值变量	距离在上一次营销活动中最后一次被联系的天数	-1 means the client was not previously contacted
15	previous	数值变量	在当次营销活动前被联系的次数	–
16	poutcome	分类变量	先前营销活动的结果	unknown, other, failure, success

（二）研究目标及评价指标

该银行获取和处理这个数据集的目的在于——通过统计模型来对新的客户进行二分类，尽可能准确而高效地预测出哪些新客户是占少数类的潜在客户（会办理定期存款，因而对银行有极高的潜在价值），其数据标签为“yes”。

但是，该数据集的正负样本很不平衡（正样本 5289 个，负样本 39922），就商业决策而言，该模型追求的不是整体准确率，而是经济效益，所以评价二元分类模型的指标应当是适应本文场景的。对于一个二元分类器，统计实际值和预测值，可以得到如表 4 所示的混淆矩阵。

表 4　二元分类器混淆矩阵

混淆矩阵	实际为 no	实际为 yes	**行和**
预测为 no	TN (True Negatives)	FN (False Negatives)	N`
预测为 yes	FP (False Positives)	TP (True Positives)	P`
列和	**N**	**P**	**-**

传统分类器关注的是整体准确率 Accuracy，希望该指标越高越好。然而，基于本文的数据背景，该银行追求的是查准率 Precision（在预测的潜在客户中，有多少真的会办理定期存款）和召回率 Recall（在实际的潜在客户中，有多少被模型找到了）。另外，基于该银行的商业需求 [1]，召回率指标会比查准率更加重要。

因此本文将利用综合指标 F-Score 来评价二元分类器，它同时考虑了 Precision 和 Recall。只有当 Precision 和 Recall 同时比较大时，F-Score 才比较大。其中，β 代表 Recall 和 Precision 的重要性偏好，β=1 时，两者重要性相同，β>1 时，Recall 重要性大于 Precision。结合表 4，以上三个指标的具体定义如下：

$$\begin{cases} \text{Recall} = TP / P, \text{其中} P = TP + FN \\ \text{Precision} = TP / P', \text{其中} P' = TP + FP \\ \text{F-Score} = \dfrac{(1+\beta^2) \times \text{Recall} \times \text{Precision}}{\beta^2 \times \text{Recall} + \text{Precision}} \end{cases}$$

（三）研究流程

我们决定以该不平衡数据为驱动，参考各种数据挖掘思想（如特征工程、数据重采样、Bagging 思想和深度学习思想），同时从数据层面和算法层面入手，旨在得到一个机器学习模型，它能尽可能准确而高效地预测出任意一批新客户中占少数类的、会办理定期存款的潜在客户。

我们将本论文的核心概括为“针对不平衡样本下二元分类问题的数据挖掘体系——银行电话营销策略研究”，该研究的具体步骤如流程图（图 1）所示。

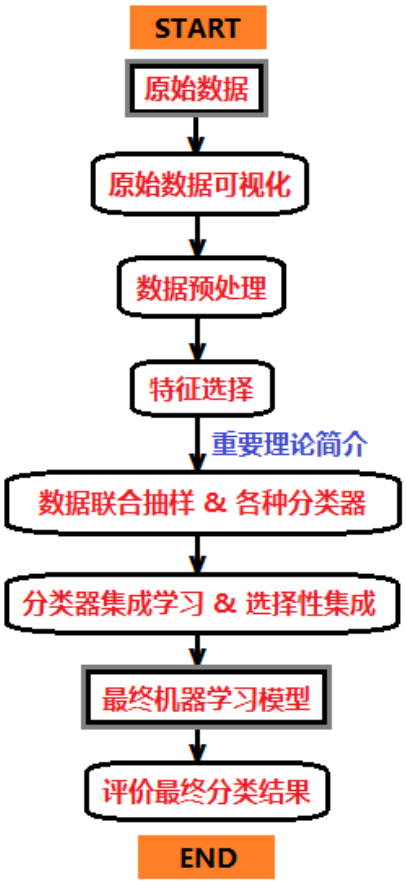

图 1　本论文基于数据挖掘的研究框架

三、原始数据可视化

该章节通过对原始数据进行可视化，有助于接下来的数据分析，为之夯实基础。下面我们从数值变量、分类变量与交叉分组三个方面来展现可视化结果。

（一）数值变量可视化

首先，我们关注该银行客户的基本信息，在删除离群点后，其年龄分布和年均存款数额（欧元）的分布如图 2 所示。我们发现，该银行客户的年龄集中分布在 30 至 60 岁，且整体偏年轻，30 至 40 岁之间的人最多；该银行客户的年均存款数额集中分布在 0 到 1000 元，且约有 8%的人存款数为 0，这些人可能会被定为睡眠客户，将面临自动注销账户的风险。

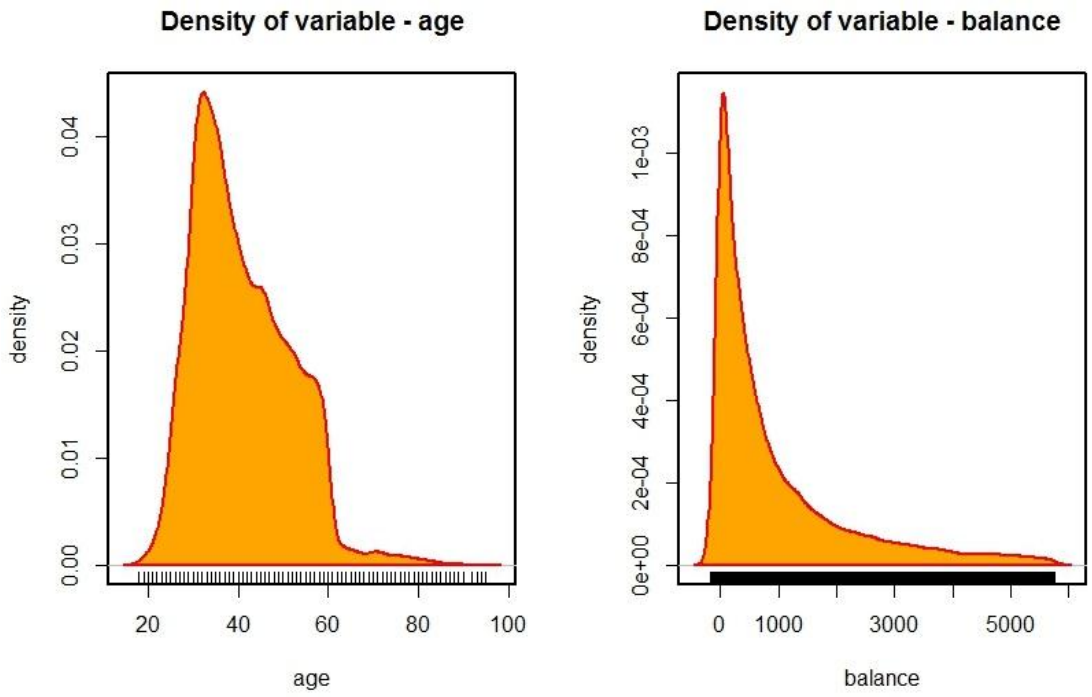

图 2　客户的年龄分布与年均存款数额分布

其次，我们关注在当次营销活动中，银行与客户的每月最后联系日期与最后电话联系时长（秒）。在删除离群点后，其箱线图如图 3 所示。我们发现，银行与大多客户在每月中旬之后当月便不再联系（中位数为 16）；最后电话联系时长主要分布在 100 到 250 秒。

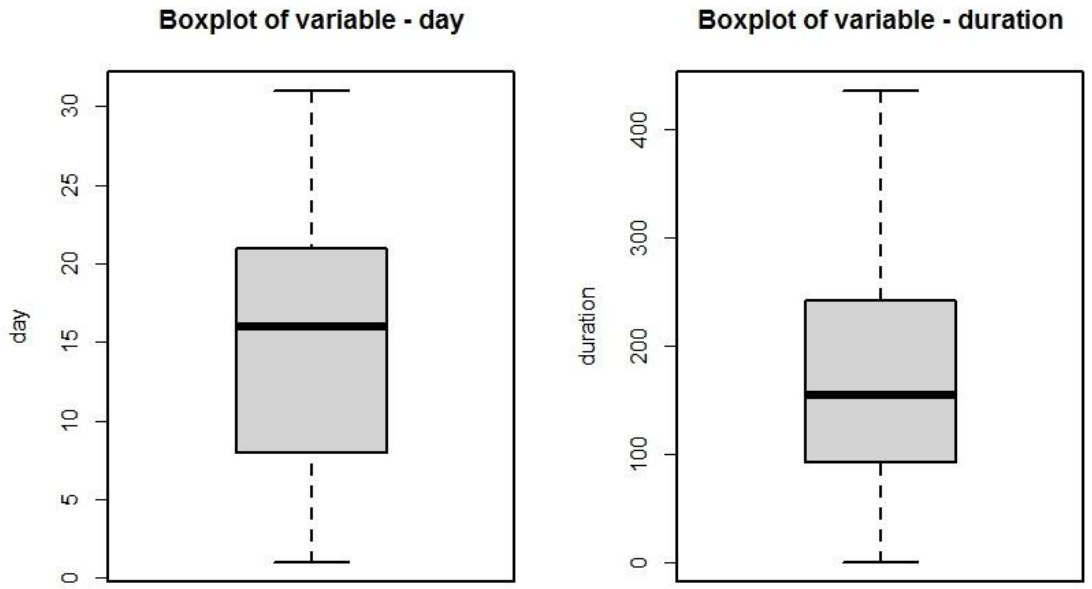

图 3　银行与客户的每月最后联系日期与最后电话联系时长（秒）– 箱线图

（二）分类变量可视化

对于分类变量本身，我们以 job 变量为例画出了条形图（图 4），便于直观地了解客户的职业群体分布。属蓝领、管理人员和技术人员的客户远多于其他类别。

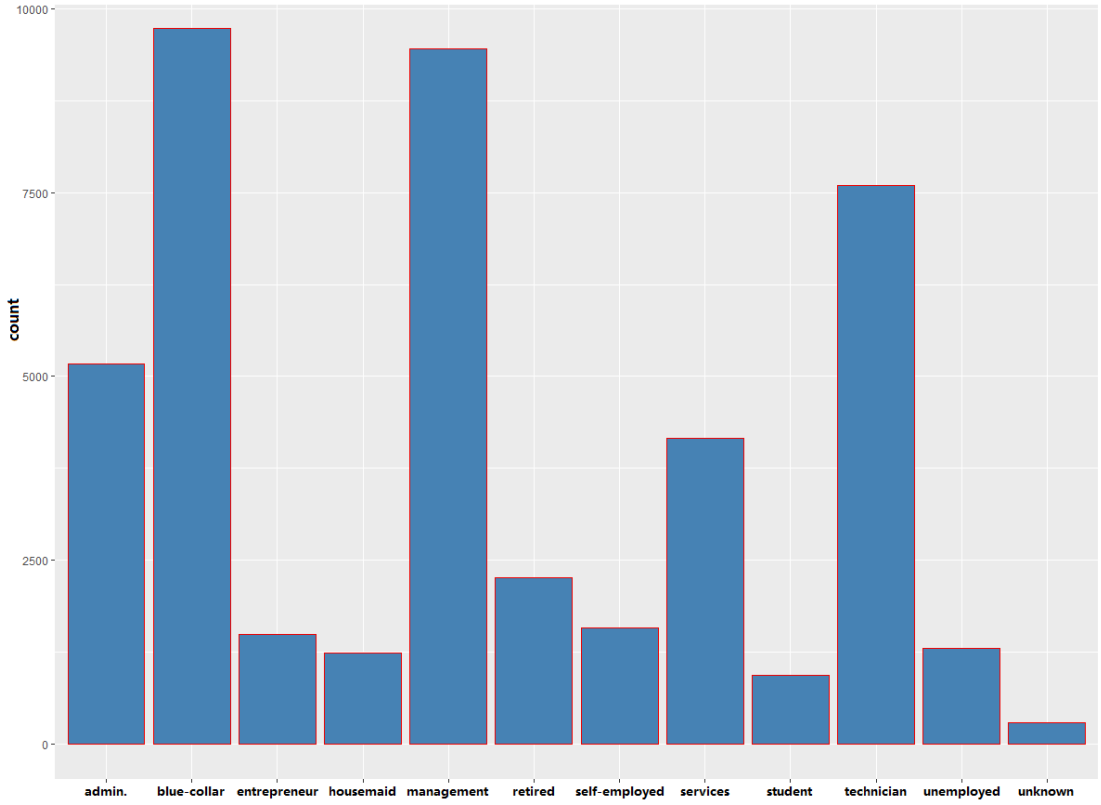

图 4　客户所源自的职业群体 – 条形图

此外，我们还关心各分类变量与目标变量 y 之间的关系，以变量 education 和 poutcome 为例，其马赛克图如图 5 所示。可以直观地看出，教育水平越高的客户，以及先前营销成功的客户，更有可能在当次营销活动中办理定期存款。

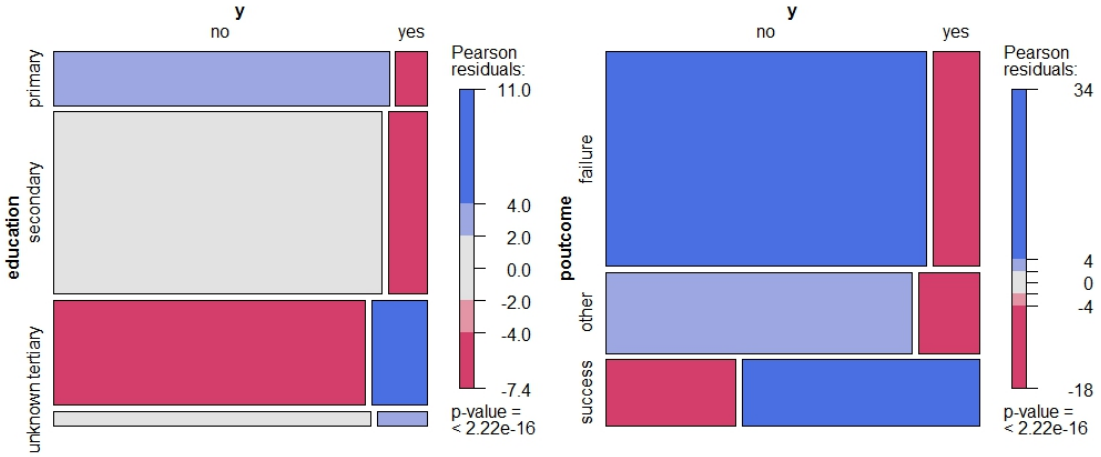

图 5　客户的教育水平、先前营销结果.与是否办理定期存款的关系 – 马赛克图

（三）交叉分组可视化

图 6 中的分组核密度图展现了不同电话沟通方式下的联系时长（秒）。

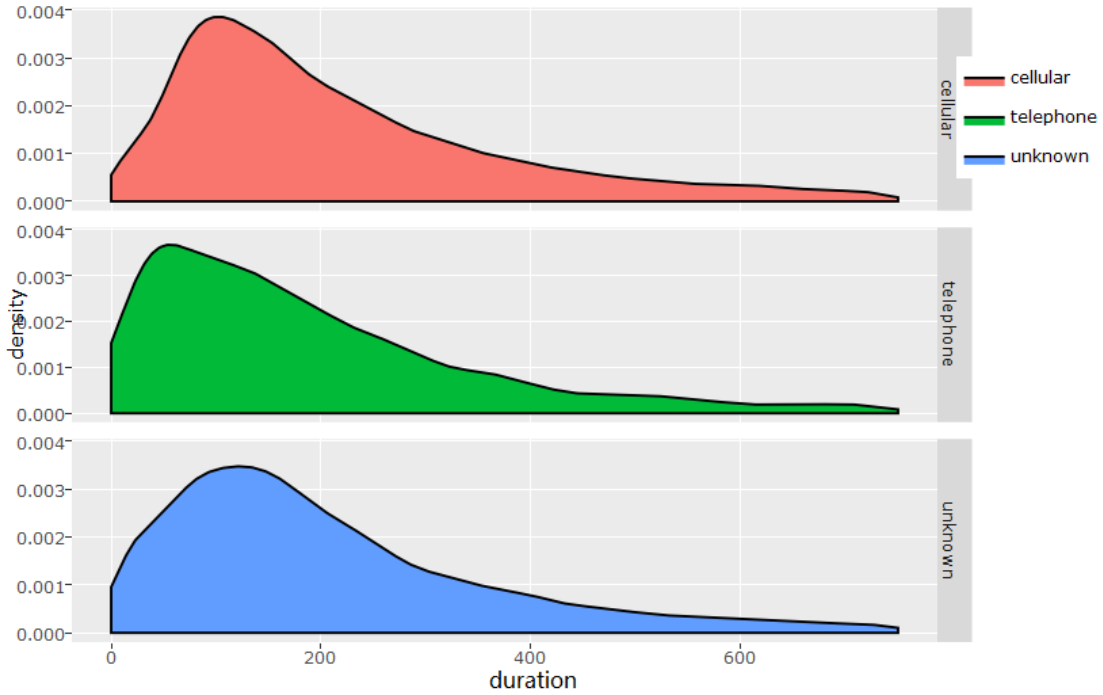

图 6　不同电话沟通方式下的联系时长（秒）– 核密度图

为了查看客户年均存款数额与信贷违约状态的关系，我们用小提琴图（核密度图与箱线图镜像叠加）来展现，详见图 7。可以看出，对于发生信贷违约的客户，其年均存款数额集中分布在 0 的右侧，明显低于信贷不违约的客户。

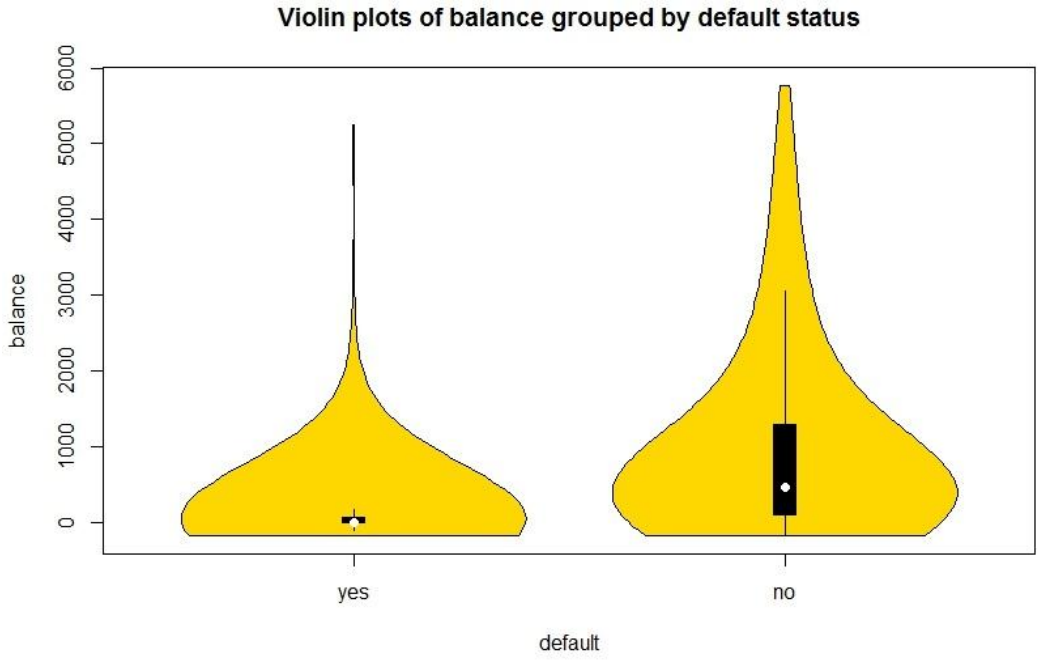

图 7　不同信贷违约状态下的客户年均存款数额分布 – 小提琴图

四、数据预处理

在原始数据 **bank-full.csv** 中，各个属性的信息采集遵从人的实际操作习惯，既包含未知信息，也涵盖了定性描述和定量描述，还有复杂变量，比较混杂。而数据和特征决定了机器学习的上限，因此，数据预处理中对不同变量需要使用相应的方法。

（一）缺失值处理

原始数据中，含有 unknown 的变量有 job、education、contact 和 poutcome，通常对含缺失值的变量会采用删除法或者填补法（如 KNN，Bagging 填补）。但对于这些变量，我们拿它和目标变量 y 在删除 unknown 值和不删除的情况下分别做 Pearson 卡方检验，发现结果显著。如果使用通常的均值填补，可能会改变样本的属性，影响后续分类模型；而删除就会浪费本来就珍贵的少数类样本。因此，我们将 unknown 作为该变量下的一个类别进行独热编码。

（二）变量处理

1. 原始数据中的二元分类变量已经是哑变量形式了（是：1；否：0），我们不做处理。（详见表 1~表 3）

2. 对于原始数据中的多元分类变量，如果它无序，我们就借助 **dummies** 包化为独热编码后的多维互斥特征[3]；如果它有序，便直接按层次顺序给每一个类别赋予单调的整数编号，类似于年龄分层变量和疾病程度变量。（详见表 1~表 3）

3. 对于复杂变量（同时包含定性记录和定量记录），先分离定性记录和定量记录，然后采用“连续变量离散化分层 + 独热编码”的组合方法来处理它，这也是本论文在数据预处理方面的创新之处。以原始数据中的 pdays 变量为例，它虽然属于数值型，却含有大量值为 - 1 的记录（这里的 - 1 不是缺失值，它缺失代表了一类具体情况，属于定性描述）。经过我们的处理，含 - 1 记录的样本被单独赋予了一个类别，并用一个新建的二元分类变量 IFP（该客户在之前的营销活动中是否被联系过）来标记它们（是：1；否：0）。

至此，数据预处理工作基本完成。重组后的新数据命名为 **bank-fullR.csv**，它包含 45211 个样本和 41 个变量。

五、特征选择

特征选择，可以提取出数据 **bank-fullR.csv** 最重要的特征，不仅能减小分类器算法的时间和空间复杂度，还能提升分类器的预测性能。

经典的特征选择方法，如主成分分析（PCA）和奇异值分解（SVD），由于存在算法适应性不强和结果解释困难的缺陷[9]，我们不予考虑。因此，我们采用目前在各项数据挖掘比赛表现优异的 Recursive Feature Elimination (RFE)、Boruta 和 XGBoost，综合三种算法挑选特征。

（一）RFE

递归特征消除（RFE）是一种寻找最优特征子集的贪心算法，它遍历探索所有可能的特征子集[16]。我们用 **caret** 包实现该方法[6]，选择出了 10 个重要特征，结果如图 8 所示。

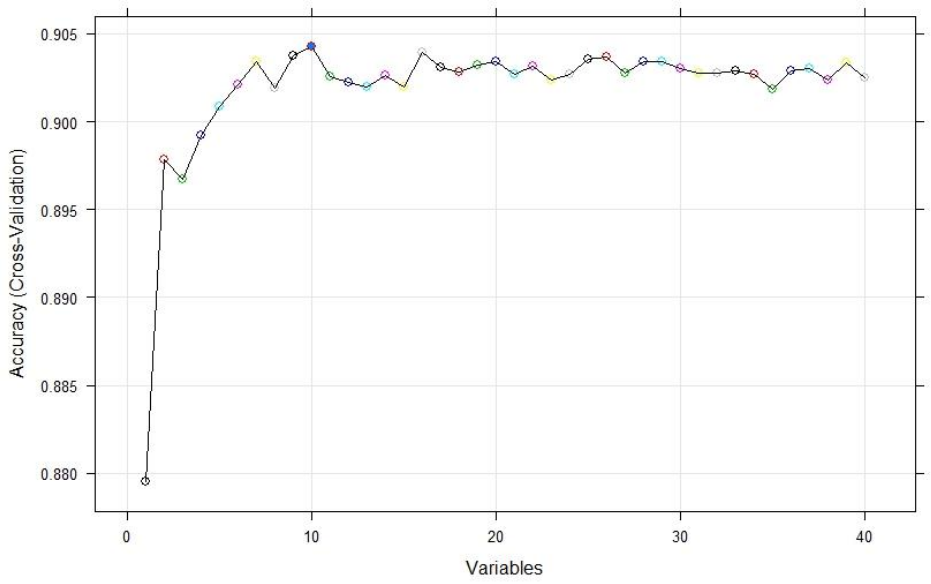

图 8　基于 RFE 特征选择方法的变量准确性（十折交叉验证）

（二）Boruta

Boruta 是一种基于随机森林的包装算法。该方法可以递归地处理每个迭代过程中表现不佳的特征，最大限度地减少模型的误差，最终形成一个最小化最优特征子集[7]。借助 **Boruta** 包[8]，我们选择出了 31 个重要特征，结果如图 9 所示。

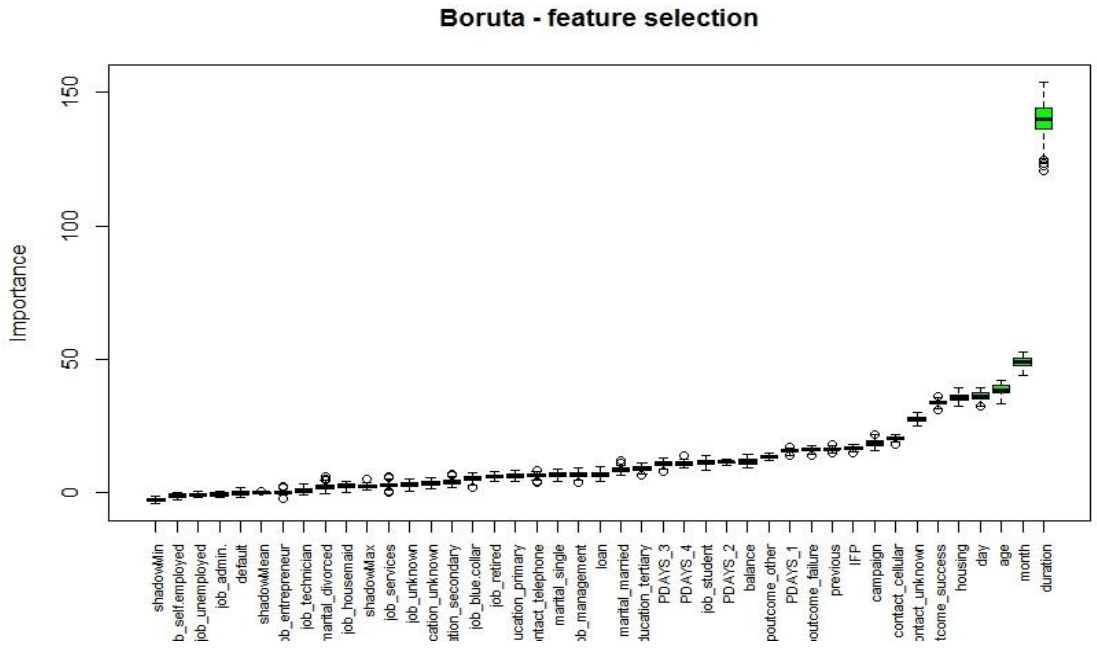

图 9　Boruta 特征选择方法下的变量重要性

（三）XGBoost

XGBoost 是对 Gradient Boosting 的实现[4]，能完成分布式计算。用 XGBoost 方法进行特征选择，其速度远超 RFE 和 Boruta 方法。利用 **xgboost** 包[5]的相关函数，经过适当调参，我们选择出了 23 个重要特征，结果如图 10 所示。

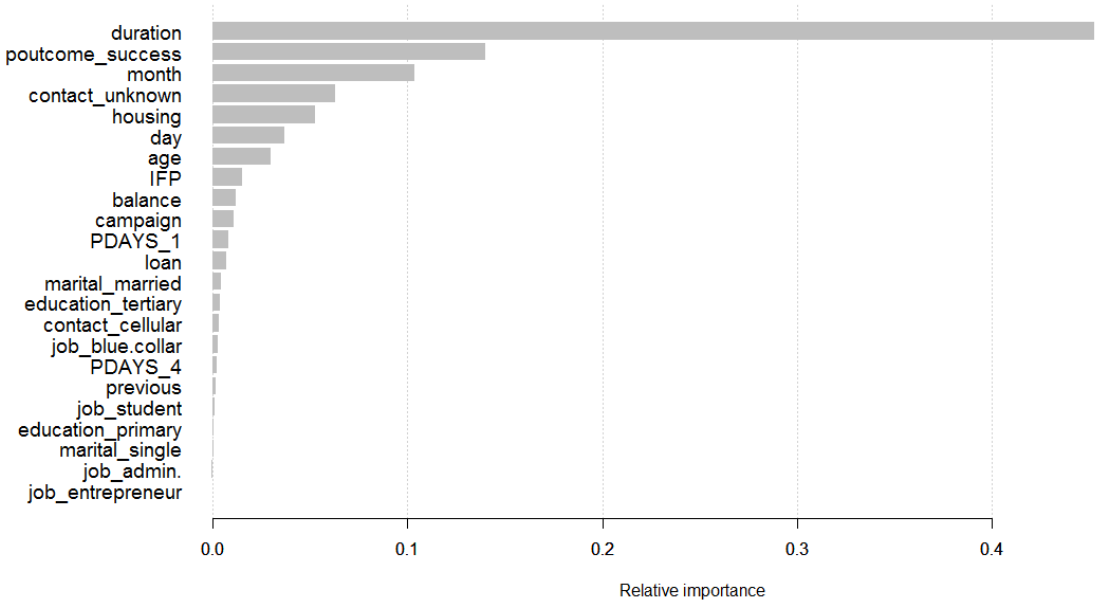

图 10　XGBoost 特征选择方法下的变量相对重要性

（四）投票选择

最后，综合以上三个特征选择方法的结果，我们进行投票选择出最终纳入考虑的特征。投票胜出的特征如下（共 21 个）：

age, day, month, duration, balance, housing, loan, campaign

education_primary, education_tertiary, marital_married, marital_single

job_blue.collar, job_student, contact_cellular, contact_unknown

IFP, previous, PDAYS_1, PDAYS_4, poutcome_success

五、经典理论简介

（一）分类器简介

本文为了提升最终的集成模型，对训练数据采用了几乎所有 R 开源的主流分类器（好的集成模型应该有多个子分类器满足相互之间有差异且准确），包括 2016 年 12 月微软刚在 R 开源的 LightGBM 算法、随机森林、ID3 决策树、C50 决策树、KNN、LDA、朴素贝叶斯、Logistic 回归、BP 神经网络、MLP 多层感知器、SVM 和 XGBoost 共 12 个分类器，限于篇幅，本文分类器大致可以划分为以下几个类别。

1. 数据挖掘早期的分类器

KNN 算法，主要思路是“近朱者赤，近墨者黑”，找到距离与待分类样本数据最近的 K 个邻居；再根据这些邻居所属的类别来判断待分类样本数据的类别。

朴素贝叶斯分类器，发源于古典数学理论，朴素贝叶斯分类器基于一个简单的假定：给定目标值时属性之间相互条件独立，在文本挖掘领域应用很广泛。

2. 经典的线性分类器

LDA 判别分析，是将高维的样本投影到最佳鉴别矢量空间，达到抽取分类信息的效果，保证投影后样本在新空间中有最小的类内距离和最大的类间距离。

Logistic 回归，是一种广义线性回归，与多重线性回归分析有很多相同之处。其区别在于 logistic 回归对因变量进行了转换。

3. 基于树模型的经典分类器

ID3 算法，是一种决策树算法，在信息论中，期望信息越小，那么信息增益

就越大，从而纯度就越高。ID3 算法的核心思想就是以信息增益来度量属性的选择。该算法采用自顶向下的贪婪搜索遍历可能的决策空间。

C50 算法，也是一种决策树，不同于 ID3 采用信息增益，C50 采用信息增益率，同时在误差估计，剪枝标准等进行了改进。

随机森林，由很多的决策树组成，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类（对于分类算法），然后看看哪一类被选择最多，就预测这个样本为那一类。

4. 复杂度高的分类器

BP 神经网络，是一种按照误差逆向传播算法训练的多层前馈神经网络，是目前应用最广泛的神经网络。从本质上讲，BP 算法就是以网络误差平方为目标函数、采用梯度下降法来计算目标函数的最小值。

MLP 多层感知器，是一种前向结构的人工神经网络，映射一组输入向量到一组输出向量。MLP 可看作是一个有向图，由多个节点层组成，除了输入节点，每个节点都是一个带有非线性激活函数的神经元。MLP 是感知器的推广，克服了感知器不能对线性不可分数据进行识别的弱点。

SVM 支持向量机，通过一个非线性映射，把样本空间映射到一个高维的特征空间中，使得空间中非线性可分的问题转化为在特征空间中的线性可分的问题。其基本模型定义为特征空间上的间隔最大的线性分类器，即学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

5. 近年数据挖掘的神兵利器

XGBoost 的全称是 eXtreme Gradient Boosting。它是 Gradient Boosting Machine 的一个 C++实现。XGBoost 最大的特点在于，它能够自动利用 CPU 的多线程进行并行，同时在算法上加以改进提高了精度。

LightGBM 是微软最新开源（发布 R 包在 2016 年 12 月）的一个基于决策树算法快速的、分布式的、高性能的 Gradient Boosting 框架，可被用于排行、分类以及其他许多机器学习任务中。

（二）不平衡数据的分类问题

不平衡数据分类困难的原因，主要有以下几点：

1. 不适合的评估指标

通常评估分类算法的指标是准确率（Accuracy），但是对于不平衡数据集来说，多数类和少数类的比例可以是 10 倍，100 倍甚至更多，少数类的分类准确率对于整体的准确率影响很小。例如数据集有 95%是多数类，5%是少数类，传统分类器可能会将所有样本全部标记为多数类，准确率可以达到 95%。但在很多实际问题中，少数类才是人们关注的。

2. 有效数据缺乏

少数类样本数量较少，导致分类器很难在少数类中发现数据规律。数据缺乏一般分为两种：绝对缺乏和相对缺乏。绝对缺乏是指少数类样本数量本身很少，故难以确定少数类的内部规律。相对缺乏是指少数类本身数量并不少，只是相对在总体的比例很小，而多数类样本太多，模糊了分类器的分类边界。

3. 噪声数据

大部分分类器的性能都会受到噪声数据的影响，而在不平衡数据中，噪声数据的影响就要超过正常数据。因为少数类样本很少，很多分类器就无法准确地辨别少数类和噪声。如果让分类器放松分类边界，那么也会意味着将噪声数据加入到了训练中，因此在真正的分类任务中可能会表现不佳。

因此，大多数分类器的分类效果可以简化为图 11 所示，多数类的准确率很高，但是少数类的准确率很低：

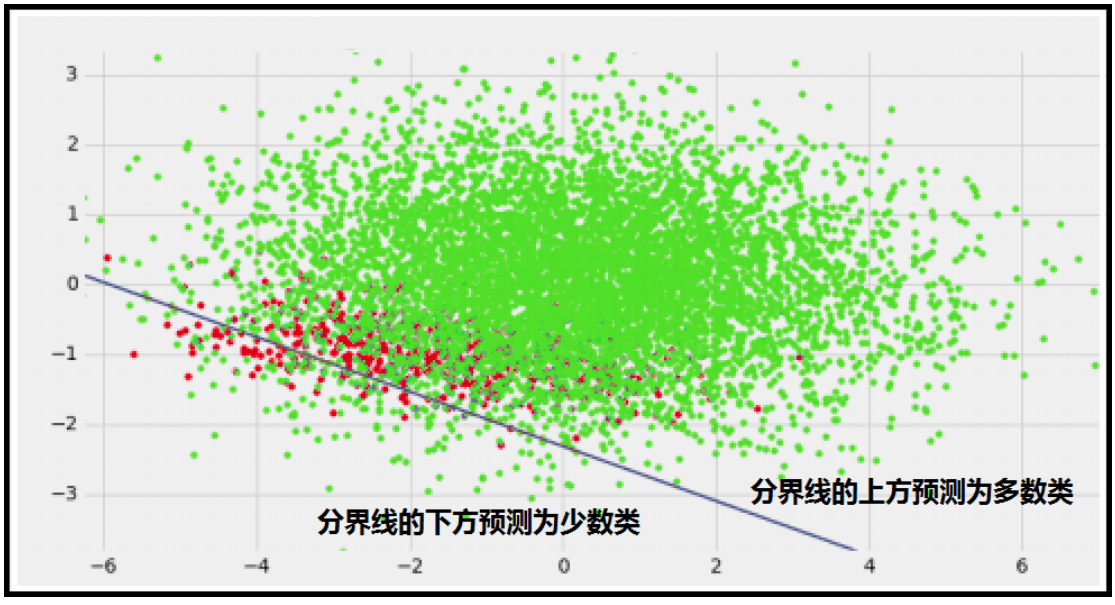

图 11　大多数分类器的分类效果

为了解决不平衡样本的分类问题，近年有较多比较成熟的理论投入了实践，本文同时从数据层面和算法层面进行研究，分别是数据重采样和分类器集成。

（三）经典的数据重采样

数据重采样，是在使用分类器算法之前，对样本数据集进行处理，减少数据不平衡性，包括欠抽样方法和过抽样方法：

1. 欠抽样方法

欠抽样方法，是指按照某种规则，删除一部分多数类中的样本，使得数据集达到整体平衡或者较平衡。

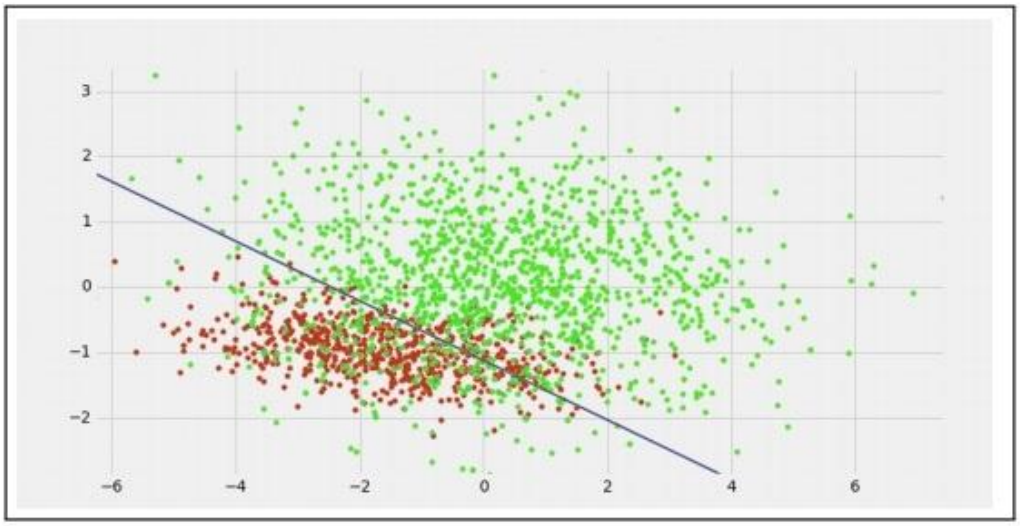

图 12　欠抽样方法下分类器的分类效果

最初的欠抽样方法是随机欠抽样，但随机删除，可能会丢失有用信息，因此，后续有学者提出了以下较成熟的欠抽样方法。

① 邻域清理法（NCL）

其主要思想是:对训练集中的每个样本，找出它的三个最近邻，如果该样本为多数类且三个最近邻中有两个以上为少数类，则视该样本为噪声样本将其删除;如果该样本为少数类，且三个最近邻中有两个以上是多数类，则将这三个最近邻中的多数类样本去掉。

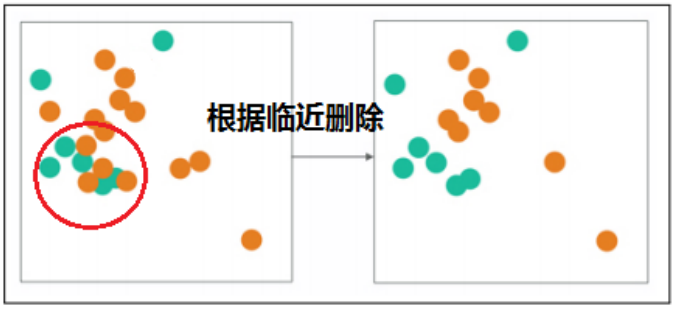

图 13　NCL 方法原理图解

② Tomek Link Removal 方法

对于两个分别属于不同类的样本 A 和 B，设数据集内找不到一个样本 C，使 A 与 C 或者 B 与 C 的距离小于 A 与 B 的距离，则样本(A,B)构成了一个 Tomek Link，其中某个样本为噪声数据，故去掉属于 Tomek Link 点的多数类样本。

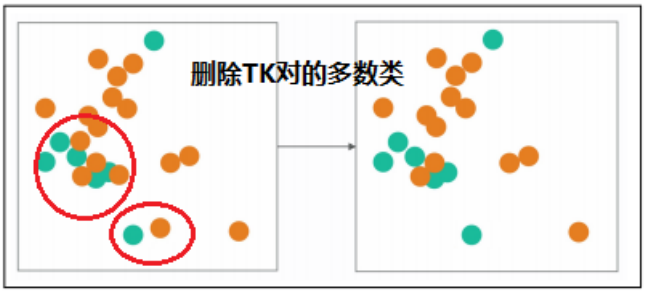

图 14　Tomek Link Removal 方法原理图解

③ K-means 聚类欠抽样

聚类欠抽样是将总体中的各个样本单位，划分成若干个互不交叉、互不重复的集合,称之为类；然后以类为抽样单位，抽取类里面的一部分样本的一种抽样方式。聚类的个数是事先给定的，可以直接影响到欠抽样的结果。

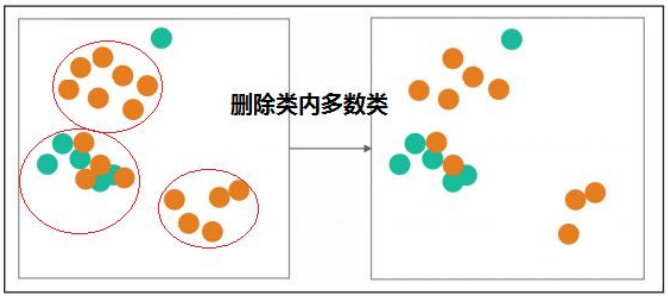

图 15　K-means 聚类欠抽样原理图解

2. 过抽样方法

过抽样方法，指通过某种规则，来增加少数类样本的数量，从而达到平衡的目的。

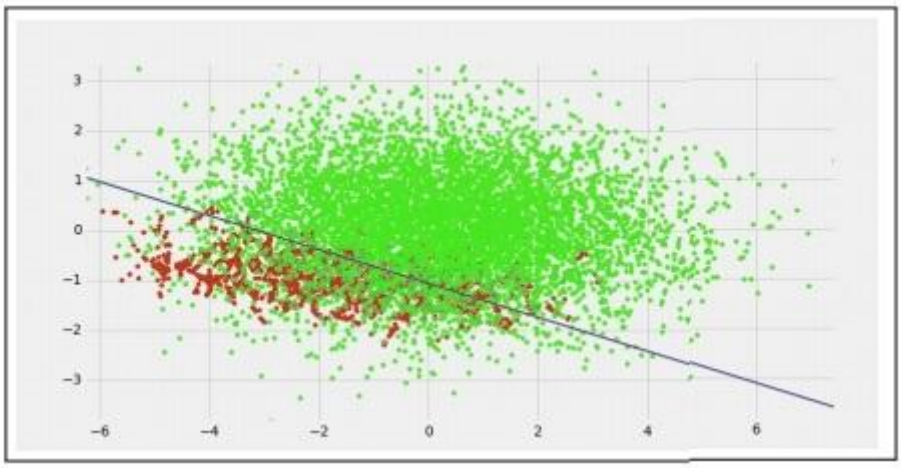

图 16　过抽样方法下分类器的分类效果

① 随机过抽样

顾名思义，随机地选择少数类样本，进行简单的复制，从而使数据集变得平衡。虽然方法简单，但是很容易造成过拟合，分类器没有学习到新的少数类信息。

② SMOTE（Synthetic Minority Oversampling Technique）

SMOTE，是过抽样中比较常用的一种方法。该算法的思想是合成新的少数类样本，合成的策略是对每个少数类样本 a，从它的最近邻中随机选一个样本 b，然后在 a、b 之间的连线上随机选一点作为新合成的少数类样本。

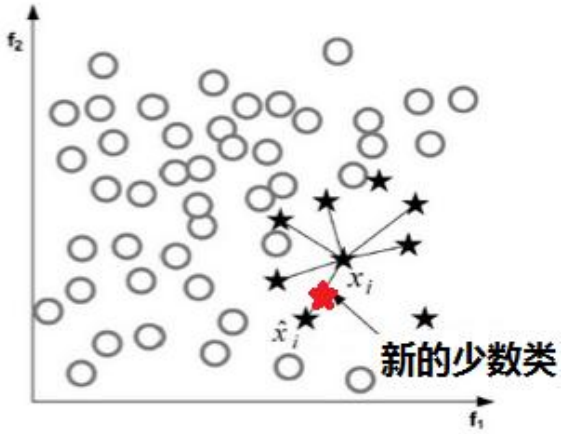

图 17　SMOTE 方法原理图解

（四）经典的分类器集成

分类器集成，是指将一系列子分类器的结果进行组合来预测样本类别。在考虑子分类器时应当尽可能地泛化能力强，差异性大。分类器集成主要有 Bagging 和 Boosting：

1. Bagging

对数据集训练多个模型，对分类问题，采用投票的方法，选择票数最多的类别作为最终的类别，而回归问题，可采用取均值的方法，取得的均值作为最终的结果。子模型采用的是数据集的子集，采用 Bootstrap（有放回的抽样方法）。

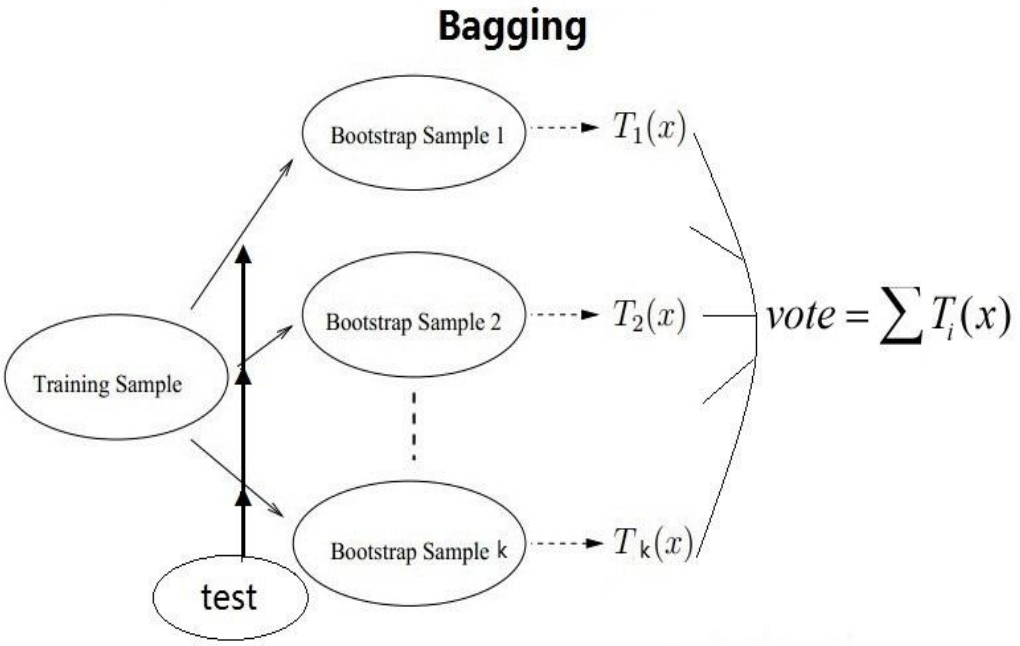

图 18　Bagging 方法原理图解

2. Boosting

初始化时对每个训练样本赋予相等的权重，如$1/n$，每次都根据前一个基分类器的表现，使预测错误的样本在之后的权重得到加强，让难学的训练样本进行重点学习。用该算法对训练集训练G轮，从而得到一个预测函数序列$\{h_1,...,h_G\}$，每个h_i都有权重，预测效果好的权重较大。最终的预测函数对多个模型的预测结果做加权平均。

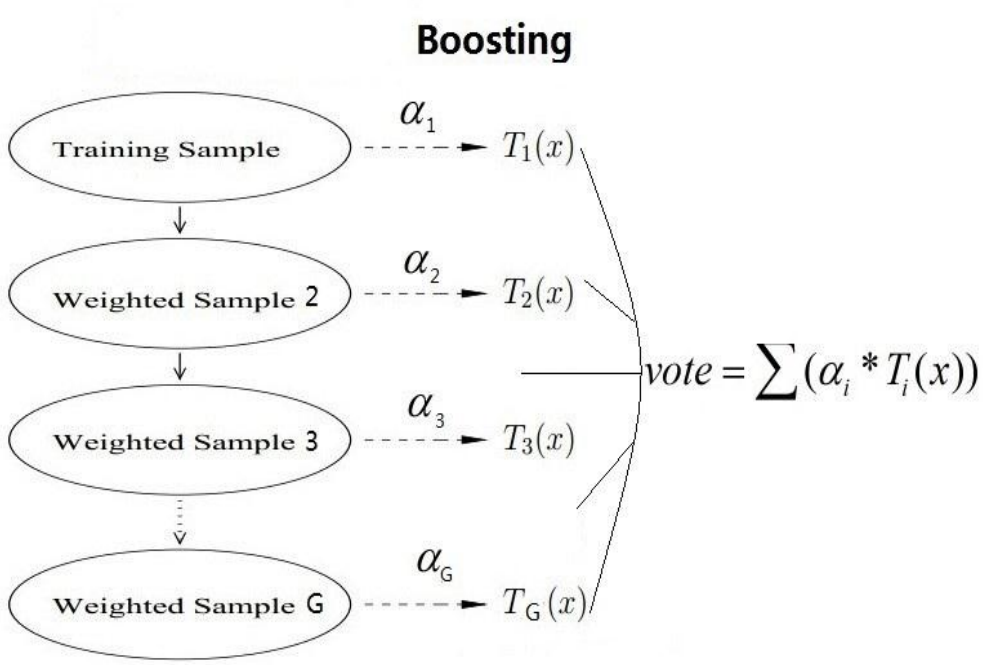

图 19　Boosting 方法原理图解

3. 选择集成

选择集成可以拥有比完全集成更好的性能，而选择集成就是在集成学习的基础上多了一个阶段，剔除冗余的分类器。可以看出，选择集成其实是一个组合优化问题，属于 NP 难问题，因此，通常有以下几种策略来寻找最优解。

① **排序法：**采用特定的评价函数，对所有基分类器进行评估，然后按照排名的次序选择分类器，排序法最大的优势在于可以快速地选择出较好的基分类器，而且该方法衍生出的选择性集成算法也很多。

② **优化法：**采用启发式搜索近似最优解，并转化为逐步优化问题，最常见的是爬山策略（Hill Climbing），前序选择策略（FSS），后序选择策略（BSS）。但该类算法通常耗费时间较长。

③ **分簇法：**一般是将基分类器进行聚类，使得相似的基分类器能够分在同一个簇，然后在每个簇的内部进行筛选，根据评估函数，剔除性能差的分类器。即保证了子分类器的差异性，也保证了准确性。

六、本文方法简介

（一）本文的联合抽样方法

在数据重采样方法中，最经典并且广泛应用的就是 SMOTE 算法了，但是 SMOTE 随机产生的新少数类样本，可能会接近多数类样本，对分类器的分类边界造成干扰。

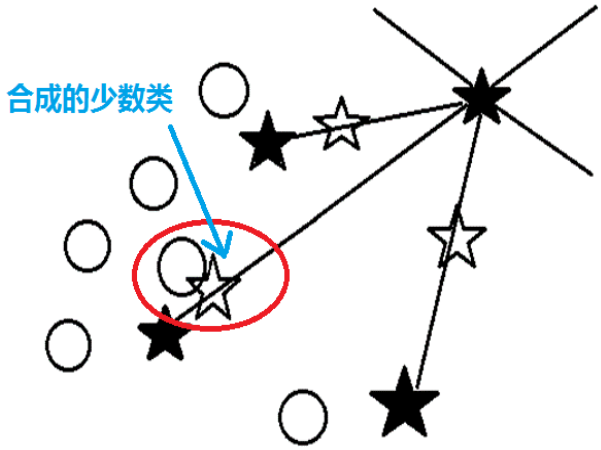

图 20　SMOTE 方法的缺陷

上述随机产生的少数类样本存在干扰样本，由于接近多数类，会被 NCL 或 Tomek Link Removal 识别出来而被删除。同时 K-means 聚类欠抽样，也能通过类内距离排序来删除上述干扰样本。因此，本文在后续实验中，采用了 SMOTE 和 NCL, Tomek Link Removal，K-means 聚类欠抽样结合的联合抽样方法。

本文采用的联合抽样方法根据算法的首字母进行命名，例如将 SMOTE 和 K-means 聚类的联合抽样方法简记为 SK，依次类推有 SNK,STK,SNT,SNTK,SNT。

总之，联合抽样方法能够同时利用欠抽样和过抽样的优势，在尽量保留多数类有效信息的情况下，也对生成的少数类进行了规则筛选，能够有效提升后续分类器的分类效果。

（二）本文的集成框架

经典集成方法是投票方法或者加权投票法，有较明显的局限性：

1. 把子分类器的结果做一个线性相加或者加权相加，得到最终的模型结果。但子分类器的非线性组合更可能接近真实值。

2. 通常是主观选择若干分类器，然后将所有分类器集成，得到预测结果。但实际上，个体基分类器的性能好，并不能保证集成效果也是好的。也不是将所有的基分类器的结果都进行集成才是最好的模型。

因此，本文提出的 XGBoost-Stacking 选择集成框架就解决了上述问题：将所有子分类器的输出当做样本的新特征，合并原始特征，然后输入到 XGBoost 中，进行第 2 层的再学习模型。该模型使用了子分类器的非线性组合，同时再次采用原特征。在第 2 层的 Stacking 再学习中，通过“分簇排序”策略来选择集成，筛选出满足相互差异性大且 F-Score 较高的子分类器对应的新特征，再使用 XGBoost 训练。具体流程如图：

1. 训练集有 10 份，反复进行留 1 预测（此时其他 9 份用做训练），直到全部 10 份训练集都有 55 个分类器下的预测值，它们被视作新特征。

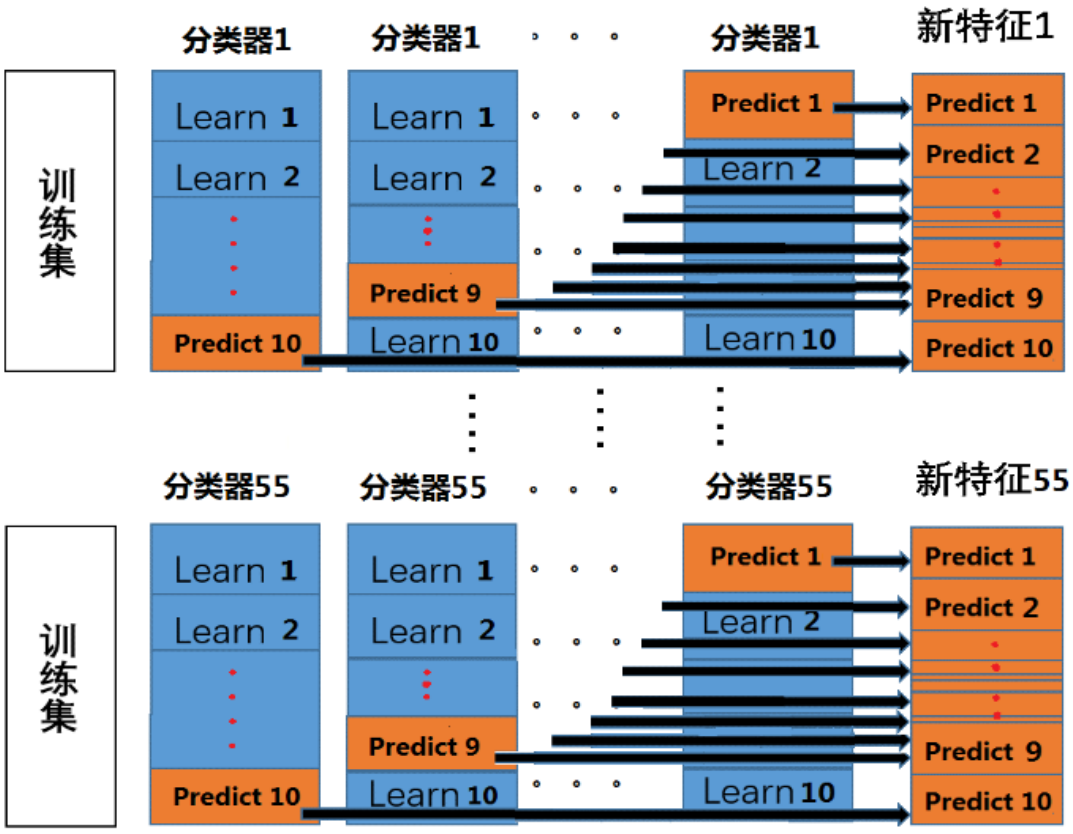

图 21　本文集成框架 – 第一步

2. 把训练得到的新特征和原始特征进行合并，加入 XGBoost 进行选择集成训练，同时可以在训练集内部得到 55 个分类器的 F-Score。

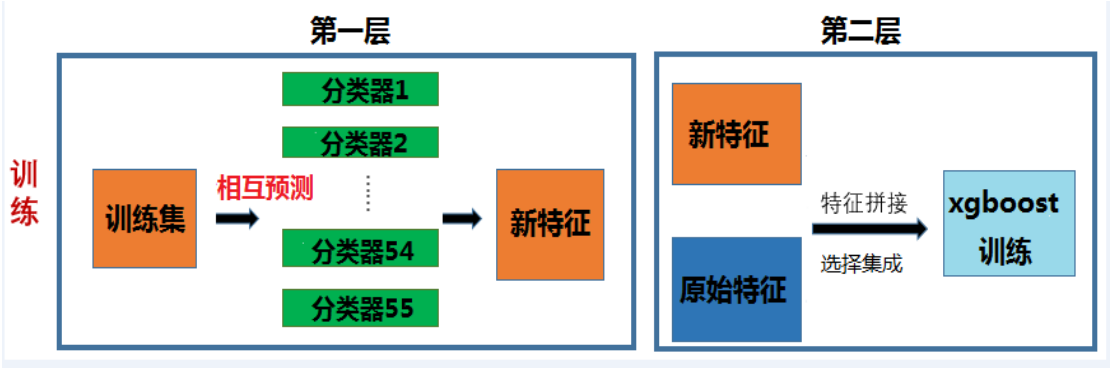

图 22　本文集成框架 – 第二步

3. 利用训练数据，对测试集进行预测，把 55 个分类器的预测特征，合并原始特征，输入到 XGBoost 进行选择集成，得到最终的分类结果。

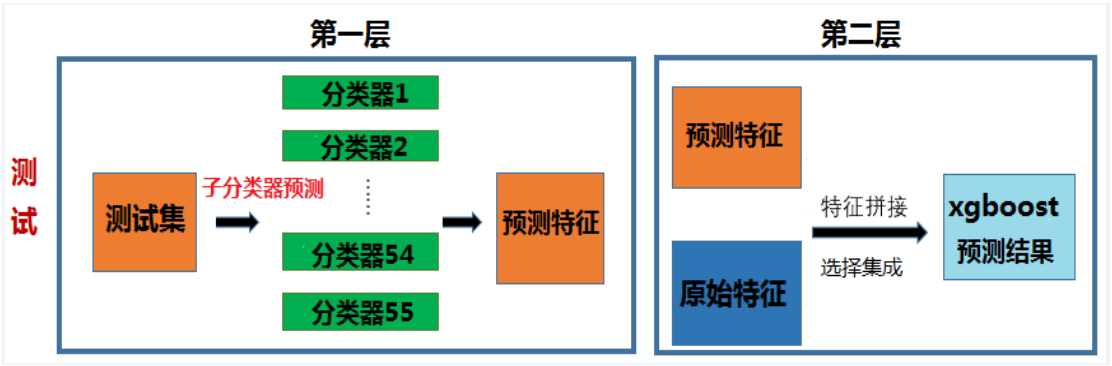

图 23　本文集成框架 – 第三步

4.选择集成即将 55 个分类器的预测作为待选新特征。再用“分簇排序”策略，根据 F-Score（步骤 2 中获得）留下每个簇里较好的子分类器结果作为最后的样本新特征。最后拼接原始特征，作为输入，由 XGBoost 输出预测结果。

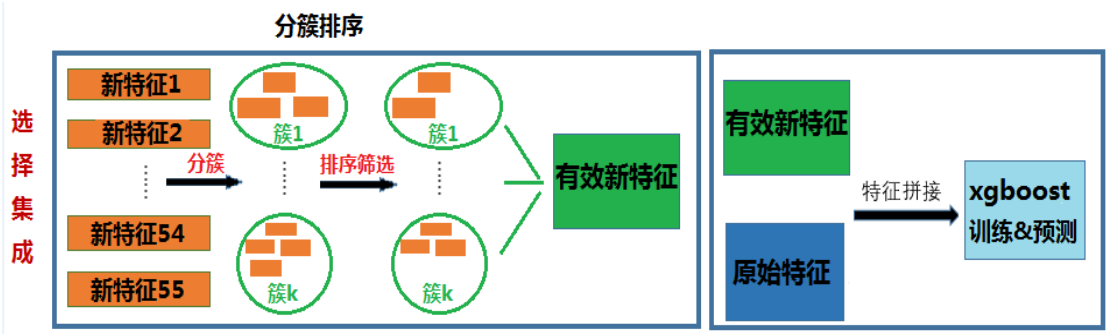

图 24　本文集成框架 – 第四步

七、模型实证

结合上一部分对本文方法的简介，我们给出模型实证的实验流程（图 24）。

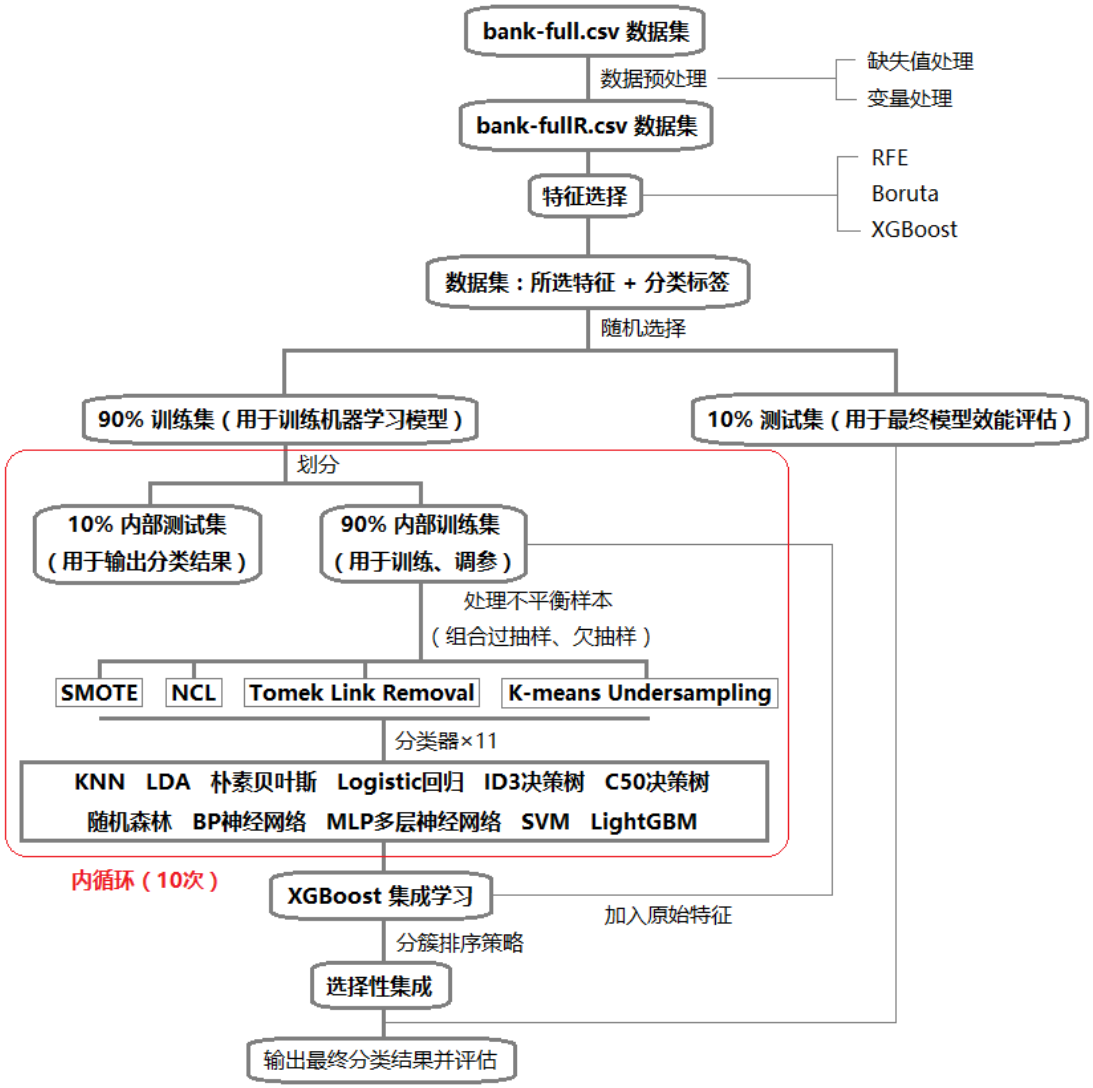

图 25　本论文的核心方法体系详解——流程图

（一）训练集和测试集

对数据集进行预处理，特征工程之后，得到一个 45211 行 22 列的数据框。首先，设置随机种子 1234，将数据集随机分割出 90%的训练集（40690 行）和 10%的测试集（4521 行）。将训练集随机平分为 10 份，用于构建本文 XGBoost-Stacking 选择集成框架。

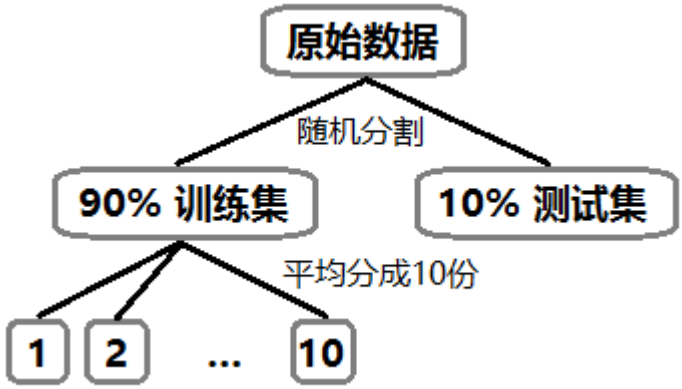

图 26　训练集和测试集的划分

（二）分类器初探

评估分类器的分类效果采用综合指标 F-Score（结合本文场景，Recall 比 Precision 更重要，令 β=1.5），对每个分类器进行初步评估，使用十折交叉验证法，得到分类器的平均表现（表 5）。

表 5　分类器的初步表现

分类器	运行时间	Precision	Recall	F-Score
KNN	94.60944	0.6496469	0.2261297	0.2828709
LDA	4.340248	0.608526	0.3832483	0.4325154
朴素贝叶斯	27.008545	0.407101	0.4465873	0.4336454
Logistic 回归	6.942397	0.6465064	0.3253923	0.3840924
ID3 决策树	25.615465	0.6336971	0.3512951	0.4071196
C50 决策树	30.37795	0.6008792	0.4910191	0.5202885
随机森林	630.765107	0.6674766	0.3635848	0.4228161
BP 神经网络	133.756734	0.6027397	0.3743619	0.4237665
MLP 多层神经网络	83.71578	0.616568	0.3940253	0.4432519
SVM	**6467.24368**	0.3181457	0.7561437	0.5311465
LightGBM	10.8	0.6087273	0.4747589	0.5092432
XGBoost	15.641895	0.4811299	0.7857818	0.657651
平均值	**627**	**0.57**	**0.44**	**0.45**

从上图可以看出，数据的不平衡特性导致分类效果普遍较差。所有分类器平均的查准率 Precision 为 0.57，而不平衡场景下最关注的召回率 Recall 只有 0.44，平均的综合评价指标 F-Score 为 0.45。在分类器耗费时间方面，SVM 的训练时间超过了 100 分钟。

（三）本文改进的 SVM（multi-SVM）

在上述分类器初探中，我们发现经典分类器 SVM 的 F-Score 能达到 0.53，超过了分类器的平均水平，但耗时太久，不利于本文模型在更大规模数据集的推广使用。因此借鉴 Bagging，我们将训练集进行分解，以下是 multi-SVM 的简要框图：

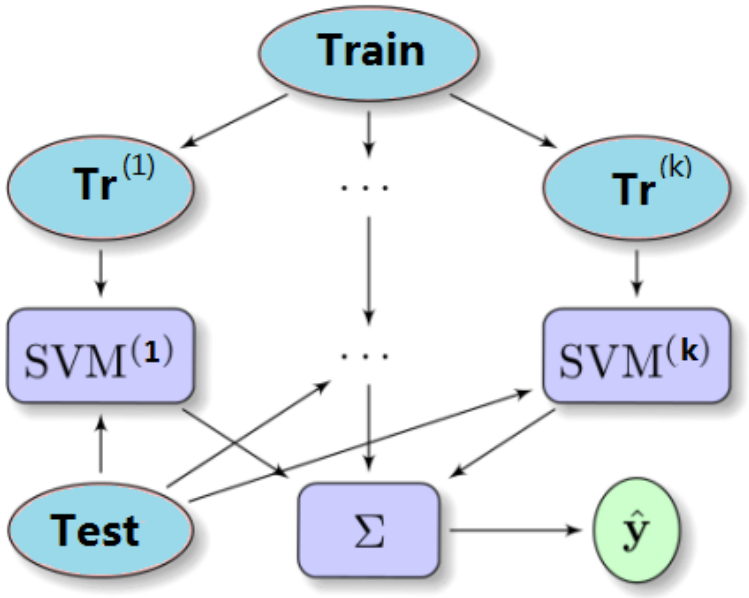

图 27　基于训练集分解方法的 multi-SVM 流程图

如图，训练了 k 个子模型 $SVM^{(1)},\ldots,SVM^{(k)}$，每个子模型的数据都是原始数据集的一个Bootstrap抽样子集。根据参考文献[40]，假设原始数据大小是 n，则在原始数据训练的复杂度是 $O(n^2)$，将数据集 n 分成 k 份，则每份数据量是 $\frac{n}{k}$，每个子模型的复杂度是 $O\left(\frac{n^2}{k^2}\right)$，合起来是 $\sum_{i=1}^{k} O\left(\frac{n^2}{k^2}\right) = O(\frac{n^2}{k})$，则复杂度比原始训练缩小了 k 倍。

本文的 multi-SVM 采用 20 个子模型，在训练数据集的表现，F-Score 为 0.5303177，与原始 SVM 几乎无差异，而时间从 107 分钟降低到了 6 分钟（理论值是 5 分钟）。

（四）数据重采样

随后我们使用了 SMOTE 和本文的 5 种联合抽样方法（SNT, SNK, STK, SK, SNTK）对 12 个分类器进行模型训练，（其中聚类的个数确定采用了 Calinsky 准则[18]和 AP 算法[17]确定，如图，聚类个数确定为 110 个）。

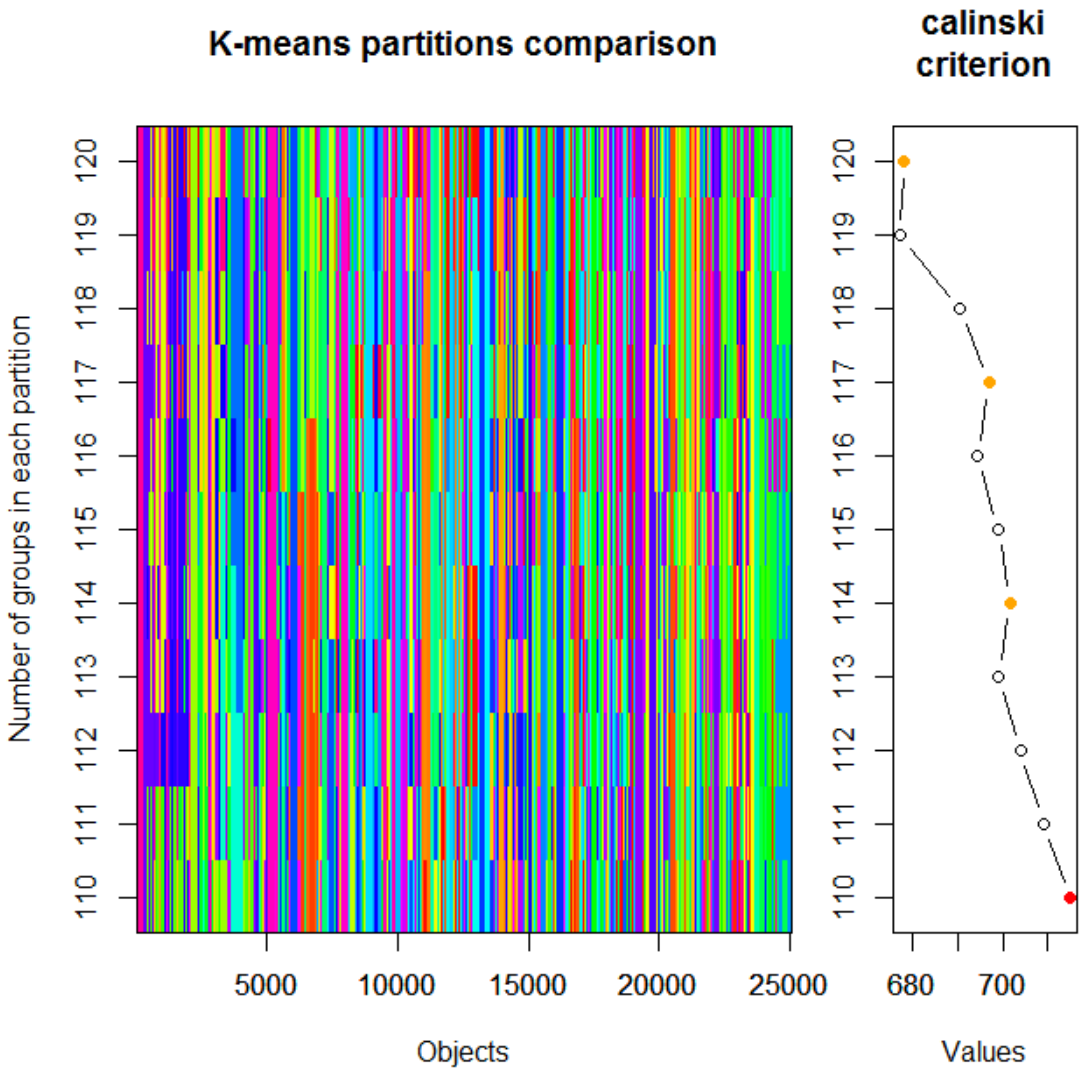

图 28　K-means 类的个数确定

我们得到 12 个分类器在各种抽样方法下的表现，如下表：

表 6　分类器表现（基于各种处理不平衡样本的方法和组合方法）

分类器\处理方法	S	SNT	SNK	STK	SK	SNTK
KNN	0.5445	0.5346	**0.5622**	0.5615	0.5602	0.5622
LDA	0.5339	0.5577	**0.6007**	0.5993	0.5994	0.5984
朴素贝叶斯	0.4697	0.5122	0.5076	0.5216	0.4942	**0.5272**
Logistic 回归	0.5602	0.5863	0.5981	**0.6011**	0.5832	0.5935
ID3 决策树	0.5648	0.559	0.5387	0.5329	**0.583**	0.5389
C50 决策树	0.6113	0.6415	0.6438	0.6446	**0.6503**	0.6412

随机森林	0.6008	0.663	0.6571	0.6604	**0.6751**	0.6541
BP 神经网络	0.5836	0.5816	0.5771	**0.5851**	0.5651	0.5765
MLP 多层神经网络	0.6094	**0.6176**	0.4929	0.493	0.5066	0.4972
multi-SVM	0.5856	**0.6016**	0.6005	0.5947	0.5128	0.5983
LightGBM	0.5925	0.6394	0.6532	0.6457	0.6461	**0.6569**
XGBoost	**0.6325**	0.6098	0.5561	0.562	0.5664	0.5564

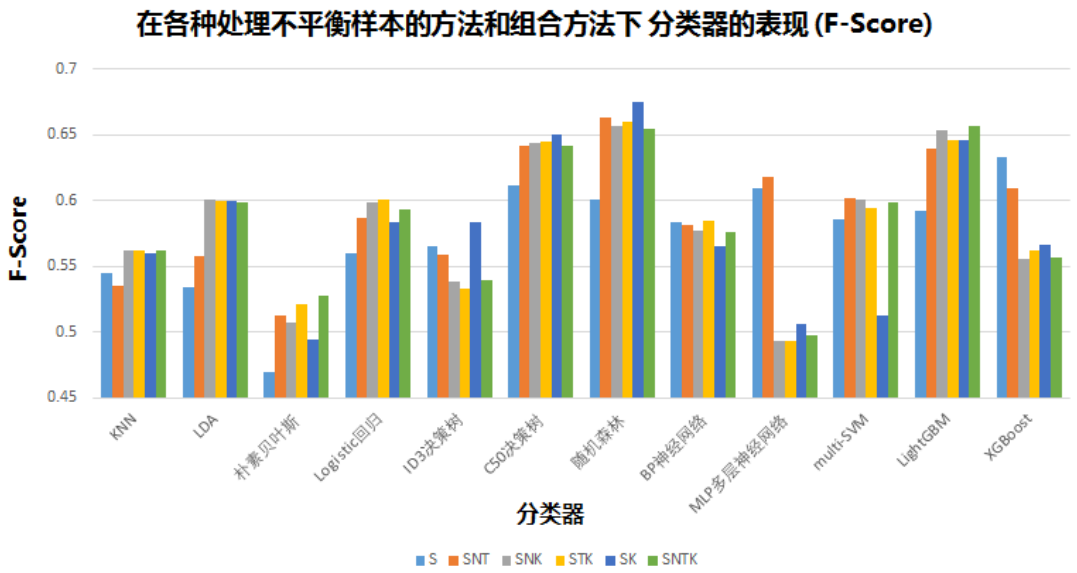

图 29　分类器表现（基于各种处理不平衡样本的方法和组合方法）

从结果来看，并结合实际经验，可知 XGBoost 对于不平衡数据已有较好的分类效果，采用数据重抽样会降低 XGBoost 的分类性能。

而其他 11 个分类器，在本文 5 种联合抽样方法下，分类性能都超越了经典的 SMOTE，但没有 1 种抽样方法可以在任何分类器上都表现最佳，为了模型的泛化能力，我们保留了 5 种抽样方法下 11 个分类器产生的 55 个子分类器。

（五）分类器集成

1. 经典的分类器集成方法

经过上述数据联合采样后，对测试集训练 55 个子分类器，在每个子分类器中，测试集都有自己的投票标签，使用以下两种普遍使用的集成方法。

① **简单投票法：** N 个分类器中投票为 1 的 n 个，若 $N \geq \frac{n}{2}$，则预测该类别为 1。

表 7　简单投票 – 混淆矩阵

混淆矩阵	实际为 no	实际为 yes
预测为 no	3353	79
预测为 yes	639	450

结果为：Precisions: 0.4132231，Recall: 0.8506616，F-Score: 0.6416584

② **加权投票法：**每个子分类器的权重取决于训练集内部训练确定的 F-Score。

表 8　加权投票 – 混淆矩阵

混淆矩阵	实际为 no	实际为 yes
预测为 no	3356	78
预测为 yes	636	451

结果为：Precisions: 0.4149034，Recall: 0.852552，F-Score: 0.6436491

2. 本文的 XGBoost-Stacking 选择集成框架

本文对训练数据平均划分为 10 份，然后训练数据中的每 1 份数据都由其他 9 份数据训练得到 55 个分类器预测值。然后通过“分簇排序”策略，将 55 个分类器结果聚类成 10 个类别，然后在每个类别里筛选留下 F-Score 最好的若干分类器。因此 10 份训练数据都拥有 41 个分类器的预测结果，原始特征，真实标签。然后把 10 份训练数据输入到 XGBoost 中进行模型再学习。最后，对于测试集，先输入到 41 个分类器，得到 41 个分类器的预测结果，然后加上原始特征，输入到 XGBoost 模型中，由 XGBoost 输出预测标签。

本文的 XGBoost-Stacking 选择集成框架的预测结果如下：

表 9　XGBoost-Stacking – 混淆矩阵

混淆矩阵	实际为 no	实际为 yes
预测为 no	3527	74
预测为 yes	465	455

结果为：Precisions: 0.4945652，Recall: 0.8601134，F-Score: 0.7007464

至此，本文的分类模型结果，能够在准确率约一半的情况下，召回率高达 86%。结合表 9 可以得出，在所有的客户样本中，我们只需要挑选约 20%的客户进行电话营销，就能够挖掘出超过 86%的潜在客户，而且这 20%的客户中，几乎一半都是潜在客户。

八、结论和意义

基于 **bank-full.csv** 数据集，本文先使用了 12 种主流的分类器进行探索，Precision 均值 0.57，而 Recall 仅有 0.44。而用本论文方法体系所搭建的机器学习模型进行二元分类时，查准率和召回率都非常高（Precision≈0.5，Recall>0.86）。在电话营销的过程中，银行只需要主动联系约 20%的客户，就能够挖掘出 86%以上具有终身价值的潜在客户，而且这 20%的客户，将近一半都是潜在客户。这一结果超过了 Sérgio Moro 与 Paulo Cortez 前辈在相似数据集上的研究结果[1]——银行只需要主动联系约 50%的客户，就能够挖掘出 79%具有终身价值的潜在客户；进一步地，上述两位前辈在同样数据集上的研究[2]采用的评价指标是 AUC 和 ALIFT，其目标性和导向性并不如本文所确立的综合指标 F-Score（同时考虑了 Precision 和 Recall）那么强。

本论文的实际意义与该银行获取这个数据集的目的基本一致[1]——构建一个用于营销与管理决策的数学模型，该银行在以后任意给定一个/一批客户，这个模型都能够准确而高效地预测出哪些新客户是占少数类的、具有一定终身价值的潜在客户。就商业决策而言，本文的分类模型系统，便于银行精确定位潜在客户（在银行办理定期存款）然后“对症下药”，就不必“大海捞针”地去试探每一个客户。可以说，依据本文模型，主动联系的客户数量比起前人成果中需要的数量降低了 60%，省下了时间和成本支出，同比还能多找到约 10%的潜在客户，提高了银行的盈利。这样的一个模型，不仅能更快速地为银行引进资金，还能减少营销成本和员工的工作压力，可以提高整个银行系统的运作效率。

本文的数据挖掘方法体系具有很强的推广性，原因有三。第一，本文是数据挖掘领域内最为常见的分类问题，F-Score 指标及分类器都可以推广到多分类情形下，因此本文的分类框架同样适应于多分类问题。第二，本文采用的分类器，多数也有回归的实现，因此本文也可以推广到回归问题。第三，本文的数据集是在各领域都很常见的不平衡数据，如金融领域的评级或风险预警；房地产，电子商务，咨询管理等诸多领域的潜在客户挖掘；广告领域的精确投放；网络领域的入侵或欺诈检测；地理气象领域的异常识别；医学领域的病人电子诊断，保险业领域的营销等等。所以，只要是对于不平衡数据下的数据挖掘问题，本文的方法和思想都值得借鉴和参考。

参考文献

[1] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. *Decision Support Systems*, Elsevier, 62: 22-31, June 2014

[2] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), *Proceedings of the European Simulation and Modelling Conference* - ESM'2011, pp. 117-121, Guimaraes, Portugal, October, 2011. EUROSIS.

[3] Christopher Brown. *Create dummy/indicator variables flexibly and efficiently*, 2012. URL http://CRAN.R-project.org/package=dummies. R package version 1.5.6.

[4] Friedman JH (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pp. 1189-1232.

[5] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang. *Extreme Gradient Boosting*, 2017. URL http://CRAN.R-project.org/package=xgboost. R package version 0.6-4.

[6] Max Kuhn. *Classification and Regression Training*, 2016. URL https://github.com/topepo/caret/. R package version 6.0-73.

[7] Miron B. Kursa, Witold R. Rudnicki. Feature Selection with the Boruta Package. *Journal of Statistical Software*, 2010, 36 (11): 1-13.

[8] Miron Bartosz Kursa, Witold Remigiusz Rudnicki. *Wrapper Algorithm for All Relevant Feature Selection*, 2017. URL https://m2.icm.edu.pl/boruta/.

[9] Peter Harrington. *Machine Learning in Action*. Manning Publications Co.: New York, 2012.

[10] Tcmek L. Two modifications of CNN [J]. IEEE *Transactions on Systems, Man, and Cybernetics*, 1976, 6: 769-772.

[11] Laurikkala J. Improving identification of difficult small classes by balancing class distribution [A]. *Proceedings of 8th Conference on AI in Medicine Europe: Artificial Intelligence Medicine*, 2001:63-66.

[12] Hart P. The condensed nearest neighbor rule [J]. IEEE *Transactions on Information Theory*, 1968, 14 (3): 515-516.

[13] Chawla N.V, Bowyer K.W. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16 (2002): 321-357.

[14] L. K. Hansen and P. Salamon. Neural network ensembles. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 1990, 12 (10): 993-1001.

[15] S. L. Sun and C. S. Zhang. Subspace ensembles for classification. *Statistical Mechanics and its Applications*, 2007, 385 (1): 199-207.

[16] Fan Li, Yiming Yang. Analysis of recursive feature elimination methods. *International Acm Sigir Conference on Research & Development in Information Retrieval*, 2005: 633-634.

[17] Ulrich Bodenhofer, Johannes Palme, Chrats Melkonian, Andreas Kothmeier. *Affinity Propagation Clustering*, 2016. URL http://www.bioinf.jku.at/software/apcluster/. R package version 1.4.3.

[18] Jari Oksanen. *Community Ecology Package*, 2017. URL https://github.com/vegandevs/vegan. R package version 2.4-3.

[19] Luis Torgo. *Functions and data for "Data Mining with R"*, 2013. URL https://CRAN.R-project.org/package=DMwR. R package version 0.4.1.

[20] David Meyer. *Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2017. URL https://CRAN.R-project.org/package=e1071. R package version 1.6-8.

[21] Gregory R. Warnes. *Various R Programming Tools for Model Fitting*, 2015. URL http://www.sf.net/projects/r-gregmisc. R package version 2.16.2.

[22] Brian Ripley. *Support Functions and Datasets for Venables and Ripley's MASS*, 2016. URL http://www.stats.ox.ac.uk/pub/MASS4/. R package version 7.3-45.

[23] Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*. 46 (3): 175–185.

[24] Brian Ripley, William Venables. *Functions for Classification*, 2015. URL http://www.stats.ox.ac.uk/pub/MASS4/. R package version 7.3-14.

[25] McLachlan, G. J. (2004). Discriminant Analysis and Statistical Pattern Recognition. *Wiley Interscience.*

[26] Hand, D. J.; Yu, K. (2001). Idiot's Bayes — not so stupid after all?. *International Statistical Review*. 69 (3): 385–399.

[27] Walker, SH; Duncan, DB (1967). Estimation of the probability of an event as a function of several independent variables. *Biometrika*. 54: 167–178.

[28] Quinlan, J. R. 1986. Induction of Decision Trees. *Mach. Learn.* 1, 1 (Mar. 1986), 81–106.

[29] Quinlan, J. R. C4.5: *Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[30] Ho, Tin Kam (1995). Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal*, QC, 14–16 August 1995. pp. 278–282.

[31] Andy Liaw. *Breiman and Cutler's Random Forests for Classification and Regression*, 2015. URL https://www.stat.berkeley.edu/~breiman/RandomForests/. R package version 4.6-12.

[32] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (8 October 1986). Learning representations by back-propagating errors. *Nature*. 323 (6088): 533–536.

[33] Brian Ripley, William Venables. *Feed-Forward Neural Networks and Multinomial Log-Linear Models*, 2016. URL http://www.stats.ox.ac.uk/pub/MASS4/. R package version 7.3-12.

[34] Rosenblatt, Frank. x. *Principles of Neurodynamics: Perceptrons and the Theory of Brain*

Mechanisms. Spartan Books, Washington DC, 1961.

[35] Christoph Bergmeir and José M. Benítez. *Neural Networks in R using the Stuttgart Neural Network Simulator (SNNS)*, 2016. URL https://github.com/cbergmeir/RSNNS. R package version 0.4-9.

[36] Vapnik, V. (1995). Support-vector networks. *Machine Learning*. 20 (3): 273–297.

[37] Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. (February 1999)

[38] Leo Breiman (1996). Bagging predictors. *Machine Learning*, 24 (2): 123–140.

[39] Leo Breiman (1996). BIAS, VARIANCE, AND ARCING CLASSIFIERS. *TECHNICAL REPORT*. Retrieved 19 January 2015.

[40] Hsieh, Cho-Jui; Chang, Kai-Wei; Lin, Chih-Jen; Keerthi, S. Sathiya; Sundararajan, S. (2008-01-01). A Dual Coordinate Descent Method for Large-scale Linear SVM. *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. New York, NY, USA: ACM: 408–415.

[41] David Meyer, Achim Zeileis, Kurt Hornik, Florian Gerber, Michael Friendly. *Visualizing Categorical Data*, 2016. URL https://CRAN.R-project.org/package=vcd. R package version 1.4-3.

[42] Carson Sievert, Chris Parmer, etc. *Create Interactive Web Graphics via 'plotly.js'*, 2017. URL https://github.com/ropensci/plotly. R package version 4.7.0

附录

附录包含许多庞大的数据表格和近两千行 R 代码，便不予在文档中展示。它们将随电子版论文，经压缩打包后一同呈递。附录文件夹名为“数据包（附录文件）”。以下为附录文件夹的内容清单和文件简介，可供查阅时参考：

附录 A　原始数据

bank.csv　随机选取 10%原始数据集样本所形成的子集（共 4521 个样本）

bank-full.csv　完整的原始数据集（共 45211 个样本）

bank-names.txt　数据集原版英文详细说明

overall description.docx　数据集原版英文简介

附录 B　中间过程数据

bank_fullR.csv　原始数据集 bank-full.csv 经过预处理之后的结果

bank_full_fs.csv　预处理数据集 bank_fullR.csv 经过特征选择之后的结果

附录 C　R 程序文件

Data Visualization.R　数据可视化章节 R 代码（77 行）

Data Pre-Processing.R　数据预处理章节 R 代码（109 行）

Feature Selection-66L.R　特征选择章节 R 代码（66 行）

part1-分类器初探.R　测试各个基分类器的初步表现 R 代码（342 行）

part2-SMOTE 抽样提升分类器.R　用 SMOTE 方法提升分类器性能（301 行）

part3-不同组合抽样方法提升分类器.R　用 5 种联合抽样方法提升分类器性能（354 行）

part4.1-stacking 框架-训练新特征.R　XGBoost-Stacking 选择集成 R 代码（356 行）

part4.2-stacking 框架-测试集结果.R　本文机器学习模型最终预测 R 代码（422 行）